ГЛАВА XX.
Линейная регрессия

Для каждой сложной задачи есть простое, понятное и неправильное решение.

Г.Л. Менкен

Змея говорила: я-то пряма, а вот щель, в какую я пролезаю, крива.

Э.М. Капиев

Предположим, что у нас есть гипотеза о линейной зависимости целевой переменной от остальных, т.е. мы ищем решение в виде

$$a(X_1,\ldots,X_n) = w_0 + w_1X_1 + \ldots + w_nX_n, \qquad \text{(XX.1)}$$

$X_1,\ldots,X_n$ – значения признаков объекта. Не всегда на практике зависимость такая простая, но предположение приводит к неплохим решениям и при монотонных зависимостях, а особенно хорошо, когда есть много «однородных» признаков (в одном масштабе и схожих по смыслу). Например, пусть

Для линейных (и метрических) алгоритмов нужны однородные признаковые пространства.

целевой признак – число продаж товара на следующей неделе,

признак 1 – число его продаж на этой неделе,

признак 2 – число заходов на страницу продукта,

признак 3 – число добавлений в корзину,

признак 4 – число появлений продукта в поисковой выдачи,

и т.п.

В этом примере все признаки отражают значения некоторых счётчиков (это неотрицательные целые числа), которые, по идее, должны коррелировать с целевым признаком.

Метод определения целевых значений по формуле (XX.1) называется «**линейной регрессией**[1]». Иногда в этой формуле не указывают **свободный член (смещение)** w_0, если есть априорные предположения о такой однородной линейной зависимости. Рассмотрим сначала линейную регрессию от одной переменной, см. рис. XX.1:

$$a(X_1) = w_0 + w_1 X_1 .$$

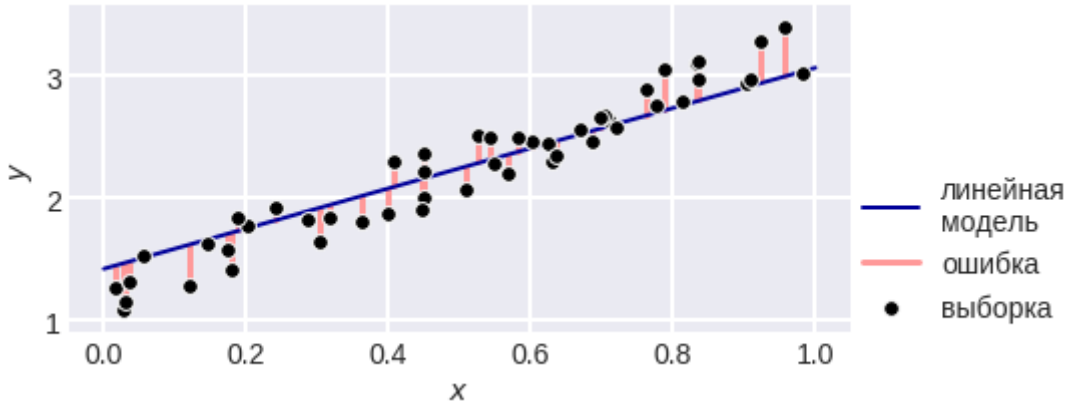


Рис. XX.1. Линейная регрессия с одним признаком.

Пусть обучающая выборка («обучение»): $\{(x_1, y_1), \ldots, (x_m, y_m)\}$, $x_i \in \mathbb{R}$, на рис.XX.1 она представляется совокупностью точек (диаграмма рассеивания или «скатерплот»). В наших «линейных предположениях» логично искать решение (коэффициенты прямой w_0, w_1) в виде решения системы уравнений

$$\begin{cases} w_0 + w_1 x_1 = y_1, \\ \cdots \\ w_0 + w_1 x_m = y_m, \end{cases}$$

но вряд ли система решается точно[2] (она может быть несовместна, особенно при довольно большом объёме обучающей выборки m), поэтому сразу перейдём к т.н. **«невязкам» (отклонениям / ошибкам / residuals)**:

$$e_1 = y_1 - w_0 + w_1 x_1,$$
$$\cdots$$
$$e_m = y_m - w_0 + w_1 x_m .$$

[1] В 1886 году Фрэнсис Гальтон исследовал зависимость роста детей и родителей и пришёл к выводу, что рост детей высоких родителей отличался от среднего роста всех детей на меньшую величину, чем рост их родителей от среднего роста всех родителей. Этот феномен был назван «возвращение к посредственности» (regression towards mediocrity), в дальнейшем термин «regression» стал использоваться для обозначения задач машинного обучения и методов из решения.

[2] Тогда все точки на рис. XX.1 просто лежат на одной прямой.

На рис. XX.1 невязки показаны розовыми отрезками. Можно поставить задачу обучения линейной регрессии как задачу **минимизации суммы квадратов отклонений (residual sum of squares)**:

$$\text{RSS} = e_1^2 + \ldots + e_m^2 \to \min .$$

На эту задачу минимизации можно смотреть как на минимизацию эмпирического риска по параметрам $w = (w_0, w_1)$:

$$L(w) = \sum_{i=1}^{m} (y_i - a_w(x_i))^2 = \sum_{i=1}^{m} (y_i - (w_0 + w_1 x_i))^2 ,$$

здесь ошибка на каждом объекте вычисляется как квадрат разности между предсказанным и истинным значениями[1]. Есть вероятностное обоснование выбора такого эмпирического риска (дальше к этому вернёмся), но пока нам достаточна внешняя логичность. Геометрический смысл ошибки показан на рис. XX.2: квадраты невязок соответствуют площадям нарисованных розовых квадратов. Линии уровня минимизируемой функции в пространстве параметров (w_0, w_1) показаны на рис. XX.2 справа. Отметим, что невязки отличаются от расстояний от точек выборки до гиперплоскости (последние используются в методе главных компонент PCA)!

Нетрудно показать (попробуйте), что решение описанной задачи минимизации RSS получается в явном виде:

$$w_1 = \frac{\sum_{i=1}^{m} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{m} (x_i - \bar{x})^2} = \frac{\text{cov}(\{x_i\}, \{y_i\})}{\text{var}(\{x_i\})},$$

$$w_0 = \bar{y} - w_1 \bar{x} \text{, где } \bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i \text{ , } \bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i .$$

Чтобы получить w_0, подставляем в уравнение $y = w_1 x + w_0$ координаты центра масс.

Получаем такое уравнение прямой:

$$(y - \bar{y}) = \frac{\text{cov}(\{x_i\}, \{y_i\})}{\text{var}(\{x_i\})}(x - \bar{x}),$$

она проходит через «центр масс» $(\bar{x}, \bar{y})$ системы точек обучающей выборки, а её коэффициент наклона зависит от ковариации $\{x_i\}$ и $\{y_i\}$.

[1] Можно минимизировать и другие ошибки, например сумму модулей разности истинных меток и предсказанных.

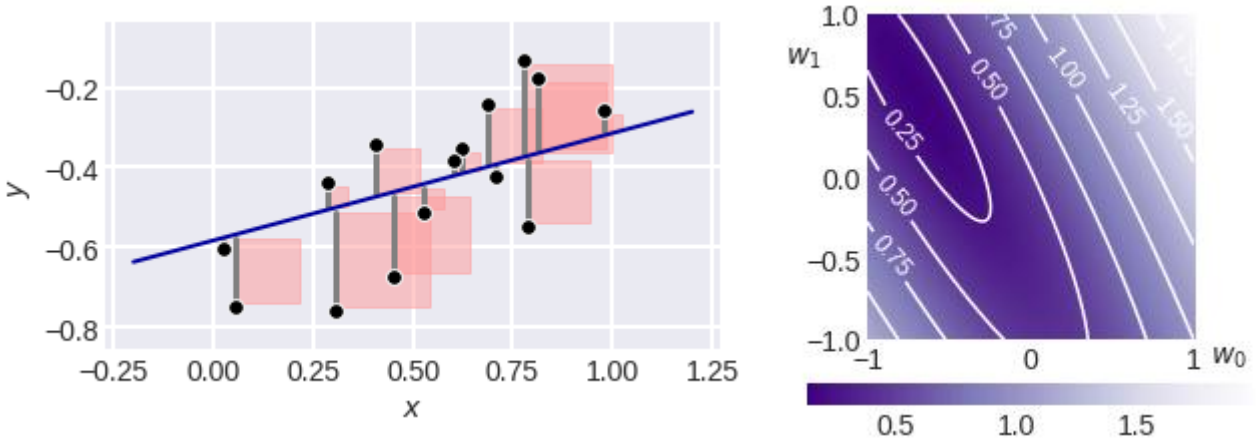


Рис. XX.2. Геометрический смысл решаемой задачи.

Линейная регрессия от многих переменных

Рассмотрим теперь общий случай (многих переменных):

$$a(X_1,\ldots,X_n) = w_0 + w_1X_1 + \cdots + w_nX_n = x^{\mathrm{T}}w,$$

где $w=(w_0,w_1,\ldots,w_n)^{\mathrm{T}}$ – вектор параметров (весов) линейной модели, $x=(X_0,X_1,\ldots,X_n)^{\mathrm{T}}$ – признаковое описание объекта. Для удобства записи мы ввели фиктивный признак $X_0 \equiv 1$ и наша линейная модель «превратилась» в скалярное произведение – в регрессию без свободного члена (смещения). Для обучающей выборки $\{(x_1,y_1),\ldots,(x_m,y_m)\}$, $x_i \in \mathbf{R}^{n+1}$, система уравнений запишется в виде

$$\begin{cases} x_1^{\mathrm{T}}w = y_1 \\ \cdots \\ x_m^{\mathrm{T}}w = y_m \end{cases}$$

или в матричной форме

$$Xw = y. \qquad \text{(XX.2)}$$

В матрице X по строкам записаны описания объектов[1], в векторе y значения их целевого признака[2]. Возникает вопрос – как решать такую систему? Мы не можем умножить на обратную матрицу к матрице X слева, и даже не потому, что матрица X может быть вырожденной, скорее всего она неквадратная

[1] т.е. это «матрица данных» (data matrix).

[2] Здесь есть коллизия в обозначении ***y*** с целевой зависимостью.

(поэтому и о вырожденности говорить некорректно). Кроме того, как и в одномерном случае, система может не иметь решения. Поэтому решаем такую задачу оптимизации

$$\| Xw - y \|_2^2 = \sum_{i=1}^{m} (x_i^{\mathrm{T}} w - y_i)^2 \to \min_w,$$

эксплуатируя уже знакомую нам идею минимизации суммы квадратов ошибок. Мы по-прежнему (как и в одномерном случае) хотим описать систему точек обучающей выборки с помощью гиперплоскости, см. рис. XX.3.

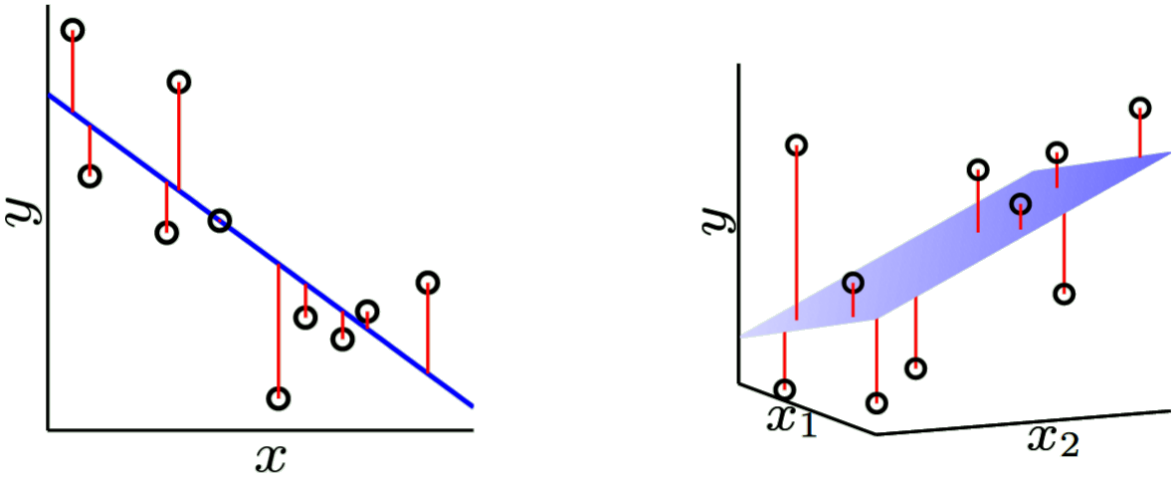


Рис. XX.3. Линейная регрессия [чужой рисунок].

Для решения задачи минимизации посчитаем градиент (по w) и приравняем его к нулю:

$$\| Xw - y \|_2^2 = (Xw - y)^{\mathrm{T}} (Xw - y) =$$

$$w^{\mathrm{T}} X^{\mathrm{T}} Xw - w^{\mathrm{T}} X^{\mathrm{T}} y - y^{\mathrm{T}} Xw + y^{\mathrm{T}} y,$$

$$\nabla \| Xw - y \|_2^2 = 2X^{\mathrm{T}} Xw - 2X^{\mathrm{T}} y = 0,$$

$$X^{\mathrm{T}} Xw = X^{\mathrm{T}} y,$$

$$w = (X^{\mathrm{T}} X)^{-1} X^{\mathrm{T}} y.$$

Правило для запоминания: исходное матричное уравнение (XX.2) умножить на X^{T} (слева и справа).

Решение существует, если столбцы матрицы X линейно независимые. Действительно, известно, что $\mathrm{rg}(X^{\mathrm{T}} X) = \mathrm{rg}(X)$, обратим также внимание, что матрица X имеет размеры $m \times (n+1)$, а матрица $X^{\mathrm{T}} X$ квадратная с размерами $(n+1) \times (n+1)$. Матрица $(X^{\mathrm{T}} X)^{-1} X^{\mathrm{T}}$ называется **псевдообратной матрицей Мура-Пенроуза** (это обобщение обратной на неквадратные матрицы).

Обобщённая линейная регрессия

Вместо матрицы X можно использовать не только признаковую матрицу, пусть мы хотим получить решение в виде разложения по т.н. **базисным функциям (basis functions)** $\varphi_1,\ldots,\varphi_k$ (они фиксированы и не зависят от данных):

$$a(X_1,\ldots,X_n) = w_0 + w_1\varphi_1(X_1,\ldots,X_n) + \cdots + w_k\varphi_k(X_1,\ldots,X_n),$$

если ввести обозначения

$$w = (w_0, w_1,\ldots,w_k)^{\mathrm{T}},$$

$$x = (X_0, X_1,\ldots,X_n)^{\mathrm{T}},$$

$$\varphi(x) = (\varphi_0(x),\varphi_1(x),\ldots,\varphi_k(x))^{\mathrm{T}},$$

где $\varphi_0(x) \equiv 1$ (константная базовая функция), то получаем

$$a(x) = \sum_{i=0}^{k} w_i\varphi_i(x) = \varphi(x)^{\mathrm{T}} w.$$

И поиск решения (коэффициентов w) свёлся к решению оптимизационной задачи

$$\| \varphi(X)w - y \|_2^2 \to \min_w$$

с матрицей

$$\varphi(X) = \begin{bmatrix} \varphi_0(x_1) & \cdots & \varphi_k(x_1) \\ \cdots & \cdots & \cdots \\ \varphi_0(x_m) & \cdots & \varphi_k(x_m) \end{bmatrix}.$$

Таким образом, в отличие от классической линейной регрессии меняется только матрица в оптимизационной задаче: $X \to \varphi(X)$. Мы как будто меняем матрицу данных на новую, переходим к новым признакам, которые порождаются базовыми функциями. В результате получается, вообще говоря, нелинейный алгоритм относительно исходных признаков, но линейный относительно новых признаков, задаваемых базисными функциями. Поэтому описанный метод решения задач регрессии получил название **обобщённой линейной регрессией**.

Проблема вырожденности матрицы

В формуле для весов (параметров) линейной регрессии

$$w = (X^{\mathrm{T}} X)^{-1} X^{\mathrm{T}} y \qquad \text{(XX.3)}$$

производится обращение матрицы, которая может оказаться вырожденной. Решения этой проблемы, с которыми мы познакомимся ниже:

- регуляризация,
- селекция (отбор) признаков,
- уменьшение размерности (в том числе, PCA),
- увеличение объёма выборки.

Заметим, что если объектов много, то работать с гигантской матрицей X размера $m \times (n+1)$, где m – число объектов, n – число признаков (к ним добавили фиктивный константный), невозможно, но можно применять онлайн-методы оптимизации (см. главу «Оптимизация»). Тем не менее, аналитическое решение (XX.3) полезно, поскольку описанные проблемы переносятся и на случай численных решений. Также обратим внимание, что на практике страшна не вырожденность матрицы (часто значения признаков заданы с некоторым шумом и из-за этого матрица невырождена), а её «близость к вырожденной», которая формализуется, например, числом обусловленности:

$$\mu(X^{\mathrm{T}} X) = \| X^{\mathrm{T}} X \| \cdot \| (X^{\mathrm{T}} X)^{-1} \| = \frac{\lambda_{\max}(X^{\mathrm{T}} X)}{\lambda_{\min}(X^{\mathrm{T}} X)}.$$

Рассмотрим первый способ решения проблемы вырожденности – **регуляризацию**. Дадим упрощённое объяснение её смысла в линейной модели

$$a(X_1, \ldots, X_n) = w_0 + w_1 X_1 + \ldots + w_n X_n,$$

очевидно, что если есть два похожих объекта, то должны быть похожи и их метки. Пусть объекты отличаются в j-м признаке на ε_j:

$$(X_1, \ldots, X_{j-1}, X_j, X_{j+1}, \ldots, X_n),$$

$$(X_1, \ldots, X_{j-1}, X_j + \varepsilon_j, X_{j+1}, \ldots, X_n),$$

тогда ответы линейной модели отличаются на $\varepsilon_j w_j$. Поэтому не должно быть аномально больших по модулю весов у признаков, по которым могут отличаться похожие объекты (подставьте в полученное произведение, скажем $\varepsilon_j = 10^{-2}$, $w_j = 10^5$), а значит и у всех признаков $X_1,\ldots,X_n$, поскольку заранее не известно, на каких объектах модель будет работать. Заметим также, что константного признака это рассуждение не касается.

Приведём ещё такой модельный пример, пусть искомая целевая зависимость

$$y = X_1,$$

т.е. целевые значения просто совпадают с первым признаком, при этом второй и третий признаки совпадают: $X_2 = X_3$ (на всех объектах обучающей выборки), тогда в линейной регрессии может получиться такой ответ:

$$a = X_1 + w'X_2 - w'X_3$$

для любого $w' \in \mathbb{R}$. Пусть теперь последние два признака не идентичны $X_2 \approx X_3$ (для какого-то нового объекта, не из обучающей выборки), как это всегда бывает на практике из-за шума / ошибок измерения значений[1]. Если на каком-то объекте значения этих признаков отличаются на шум $\varepsilon = X_2 - X_3$, то ответ

$$a = X_1 + w'\varepsilon$$

может сколь угодно отличаться от истинного $a = X_1$ при больших по модулю значениях w'. Этот пример легко обобщается на случай, когда в признаках есть линейные зависимости (здесь мы рассмотрели только равенство признаков). В эконометрике наличие таких зависимостей называется **мультиколлинеарностью (multicollinearity)**.

Заметим, что в приведённых примерах, нет причин требовать «небольших по модулю значений» от коэффициента w_0. Кроме того, по смыслу он отвечает за среднее целевое значение (если среднее всех признаков равно нулю, а так часто бывает после стандартизации, см. дальше).

Поэтому вместе с задачей $\|Xw - y\|_2^2 \to \min$ обычно стараются решить задачу $\|w\|_2^2 \to \min$ (минимизации нормы весов), где $\|w\|_2^2 = w_1^2 + w_2^2 + \ldots + w_n^2$ (здесь мы

[1] Например, в тестовой выборке появился шум из-за чего эти признаки стали различаться.

как раз не включили вес w_0). Есть следующие стандартные способы «совмещения» этих задач: **регуляризация Иванова** –

$$\begin{cases} \| Xw - y \|_2^2 \to \min \\ \| w \|_2^2 \le \lambda, \end{cases}$$

регуляризация Тихонова –

$$\| Xw - y \|_2^2 + \lambda \| w \|_2^2 \to \min, \ \lambda \ge 0,$$

вторая чуть удобнее тем, что здесь безусловная оптимизация[1]. Эти две формы эквивалентны: решение одного можно получить как решение другого (возможно, при другом значении λ). Описанные регуляризации применяются в машинном обучении и в общем случае:

$$\begin{cases} L(a) \to \min \\ \text{complexity}(a) \le \lambda \end{cases} \qquad L(a) + \lambda \, \text{complexity}(a) \to \min,$$

здесь L – эмпирический риск, complexity – функция отражающая «сложность модели» (дальше поймём, почему норма вектора коэффициентов линейной регрессии связана со сложностью модели), минимизация производится по параметрам модели.

Решение указанной задачи регуляризации Тихонова задаётся формулой

$$\arg\min \| Xw - y \|_2^2 + \lambda \| w \|_2^2 = (X^{\mathrm{T}} X + \lambda I)^{-1} X^{\mathrm{T}} y. \qquad \text{(XX.4)}$$

Для доказательства достаточно взять градиент и приравнять к нулю, как мы это делали раньше (доказать!). Регрессия с коэффициентами, определяемыми формулой (XX.4) называется **гребневой регрессией (Ridge Regression)**. Такой вид регуляризации называется также **L2-регуляризацией**, т.к. в регуляризационной добавке стоит квадрат L2-нормы вектора параметров, эту добавку мы также будем называть **регуляризатором** и **L2-штрафом**. Поясним, почему гребневая регрессия помогает в борьбе с вырожденностью (плохой обусловленностью) матрицы $X^{\mathrm{T}} X$. К неотрицательно определённой матрице Грама $X^{\mathrm{T}} X$ мы при $\lambda > 0$ прибавляем положительно определённую диагональную матрицу Грама[2]. В итоге получается заведомо невырожденная

[1] Есть ещё регуляризация Морозова... нетрудно догадаться, что в ней ограничивают эмпирический риск и минимизируют квадрат нормы вектора весов.

[2] У неё положительные элементы на диагонали как бы образуют «гребень», но термин гребневая регрессия (Ridge Regression) возник не из-за этого, до её появления был гребневый анализ (Ridge Analysis) и тут отсылка к

матрица (т.к. она положительно определена). На рис. XX.4 показана поверхность минимизируемой функции в линейной регрессии (слева) при наличии мультиколлинеарности: нет единственного решения задачи оптимизации, и в гребневой регрессии (справа): есть глобальный минимум.

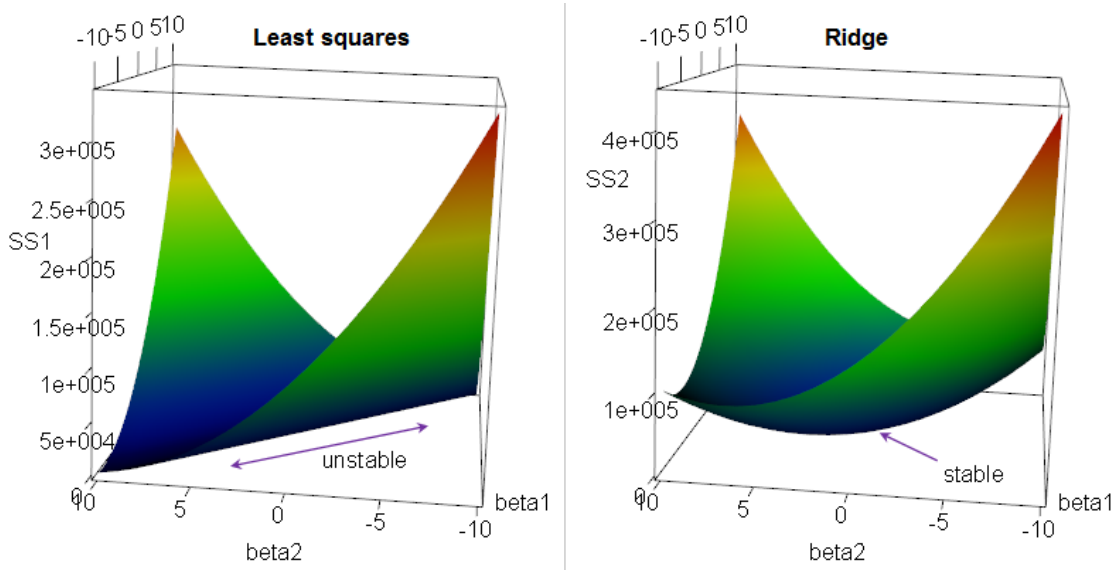


Рис. XX.4. Отличия в минимизируемых функциях: линейная и гребневая регрессии [чужой рисунок].

При $\lambda = 0$ в (XX.4) получаем классическое решение и вынуждены обращать матрицу $X^{\mathrm{T}}X$, при $\lambda \to +\infty$ матрица, которую приходится обращать, становится заведомо хорошо обусловленной, но метод меньше «затачивается на данные». Коэффициент λ называется **коэффициентом регуляризации (shrinkage penalty)**. На рис. XX.5 показан эффект от изменения коэффициента регуляризации[1]: при больших коэффициентах параметр, отвечающий за наклон прямой, становится близким к нулю и наше решение практически превращается в константное (оно соответствует среднему целевому значению). Это вполне логично, при $\lambda \to +\infty$ мы «заставляем» все коэффициенты занулиться, кроме свободного члена w_0, и наше решение превращается в константное

$$a(X_1,\ldots,X_n) = w_0 .$$

форме поверхности функций, с которыми работали, см. рис. XX.4. Hoerl R. W. Ridge regression: a historical context //Technometrics. – 2020. – Т. 62. – №. 4. – С. 420-425.

[1] Визуально кажется, что решение отклоняется к выбросам, но дело не в них.

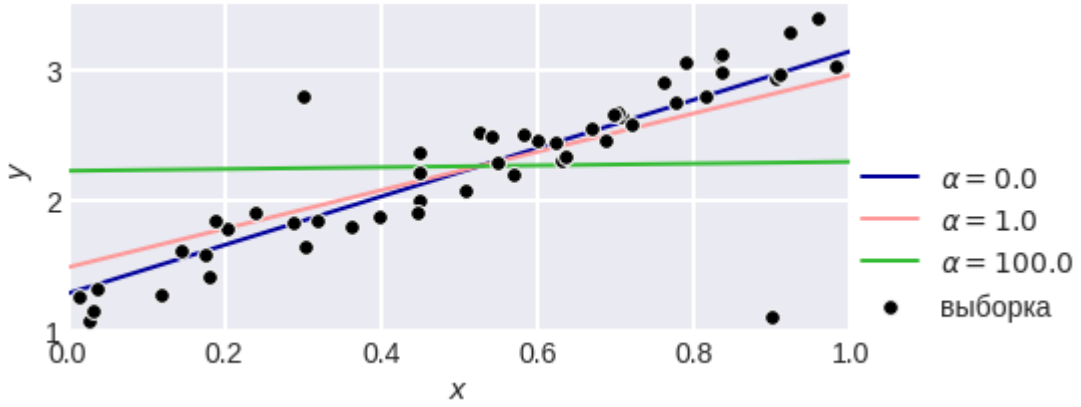


Рис. XX.5. Результаты регуляризации при различных значениях коэффициента регуляризации.

Покажем, как решение (получаемые коэффициенты) задачи ridge-регрессии

$$\sum_{i=1}^{m}\left(y_i - a(x_i)\right)^2 + \lambda\sum_{j=1}^{n} w_j^2 \to \min \qquad \text{(XX.5)}$$

(это просто другая форма записи выражения, стоящего в (XX.4)) зависит от коэффициента регуляризации $\lambda \geq 0$, см. рис. XX.6. Здесь и далее на рисунках коэффициент регуляризации обозначается через α, поскольку в стандартных библиотеках он чаще называется «alpha», а в формулах принято использовать обозначение λ. При увеличении коэффициента параметры линейной регрессии устремляются к нулю (заметим что здесь ни один параметр не занулился). Параметр регуляризации λ может подбираться с помощью скользящего контроля (см. главу «Контроль»).

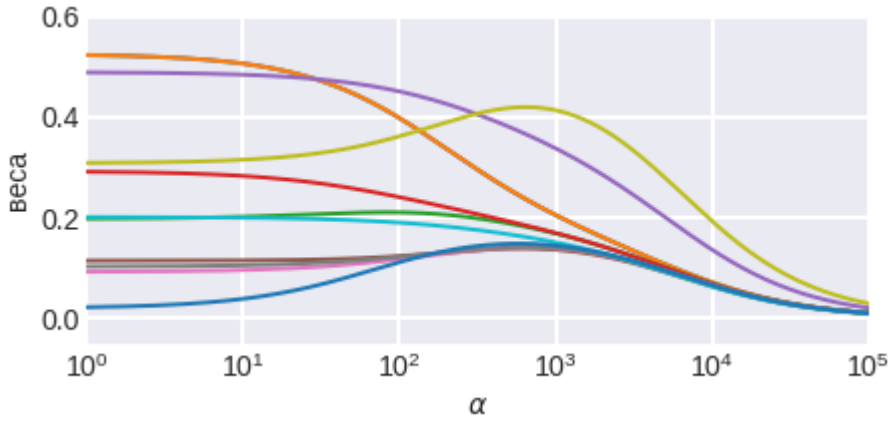


Рис. XX.6. Коэффициенты гребневой регрессии от коэффициента регуляризации.

Отметим, что для ridge-регрессии нужна правильная нормировка признаков (как правило, стандартизация), при масштабировании (умножении признаков на

скаляры) результат может отличаться[1]. Это можно проиллюстрировать в двумерном случае:

$$\sum_{i=1}^{m}\left(y_i - w_1 X_1 - w_2 X_2 - w_0\right)^2 + \lambda w_1^2 + \lambda w_2^2 \to \min,$$

если первый признак «уменьшить в k раз»: $X_1 \to X_1 / k$, то если бы не было регуляризации, коэффициент w_1 просто увеличился бы в k раз, с регуляризацией же при таком увеличении вырастает и регуляризационный штраф: $\lambda w_1^2 \to \lambda k^2 w_1^2$, поэтому в задаче оптимизации получится, вообще говоря, другое решение.

LASSO (Least Absolute Shrinkage and Selection Operator)

Аналогично покажем изменение решения (коэффициентов) в задаче

$$\sum_{i=1}^{m}\left(y_i - a(x_i)\right)^2 + \lambda \sum_{j=1}^{n} |\, w_j \,| \to \min$$

в зависимости от параметра $\lambda \geq 0$. Решение этой задачи используется в методе LASSO, здесь используется **L1-регуляризация** (в регуляризационной добавке L1-норма вектора параметров). и веса интенсивнее зануляются при увеличении коэффициента регуляризации, см. рис. XX.7. Также обратим внимание, что основные изменения коэффициентов происходят в другом диапазоне (на несколько порядков меньше).

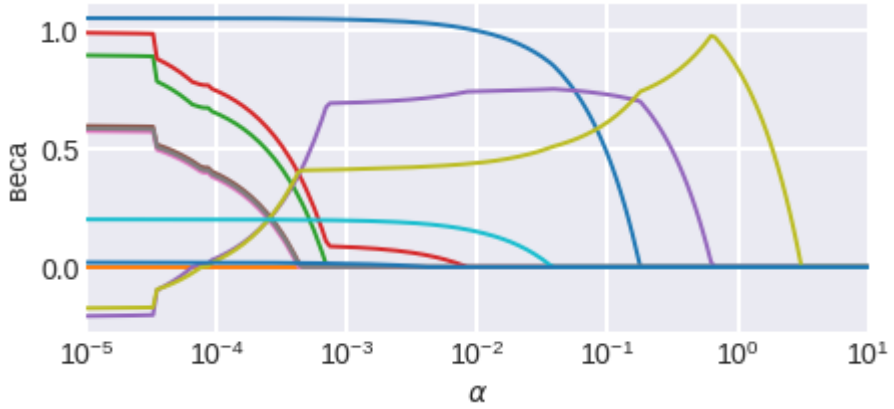


Рис. XX.7. Коэффициенты LASSO от коэффициента регуляризации.

[1] Этого не происходит в линейной регрессии (без регуляризации).

Обратим внимание, что в задаче с двумя одинаковыми признаками $X_1=X_2$, в которой целевое значение $Y=4X_1$ формально подходит решение вида

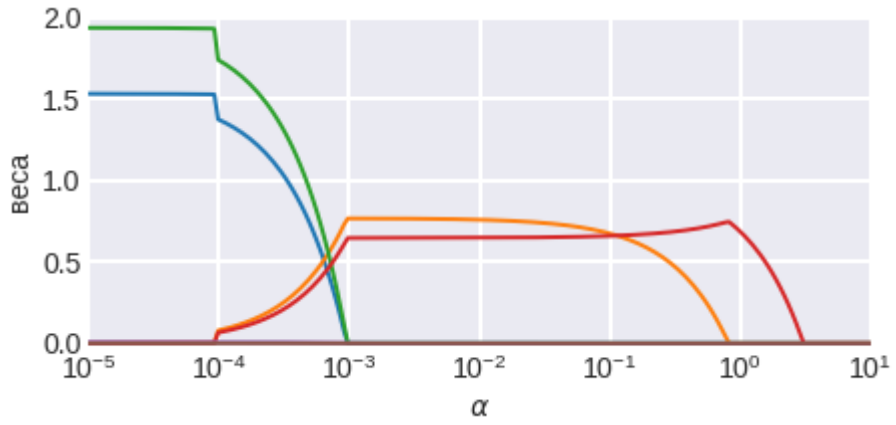
$$a=w_1X_1+(4-w_1)X_2.$$

и L2-штраф $\|w\|_2^2=w_1^2+(4-w_1)^2$ имеет единственный минимум в точке $w_1=2$, т.е.

$$a(X_1,X_2)=2X_1+2X_2$$

(веса равномерно распределяются по признакам), L1-штраф имеет вид $\|w\|_1=|w_1|+|4-w_1|$ и минимум достигается при любых $w_1\in[0,4]$, например при $a=4X_2$.

На рис. XX.8 показано, как меняются коэффициенты в одной модельной задаче при использовании свободного члена[1] w_0 и стандартизации исходных признаков[2]. Видно, что при стандартизации и свободном члене графики становятся гладкими и меньше значащих (ненулевых) параметров. На практике всегда лучше использовать описанные приёмы (кроме случая, когда из априорных соображений у нас однородная линейная зависимость – без смещения, тогда сразу полагаем $w_0=0$).



[1] В реализации sklearn это параметр fit_intercept=True.

[2] В реализации sklearn раньше это был параметр normalize=True, теперь нормализация вынесена в отдельный модуль.

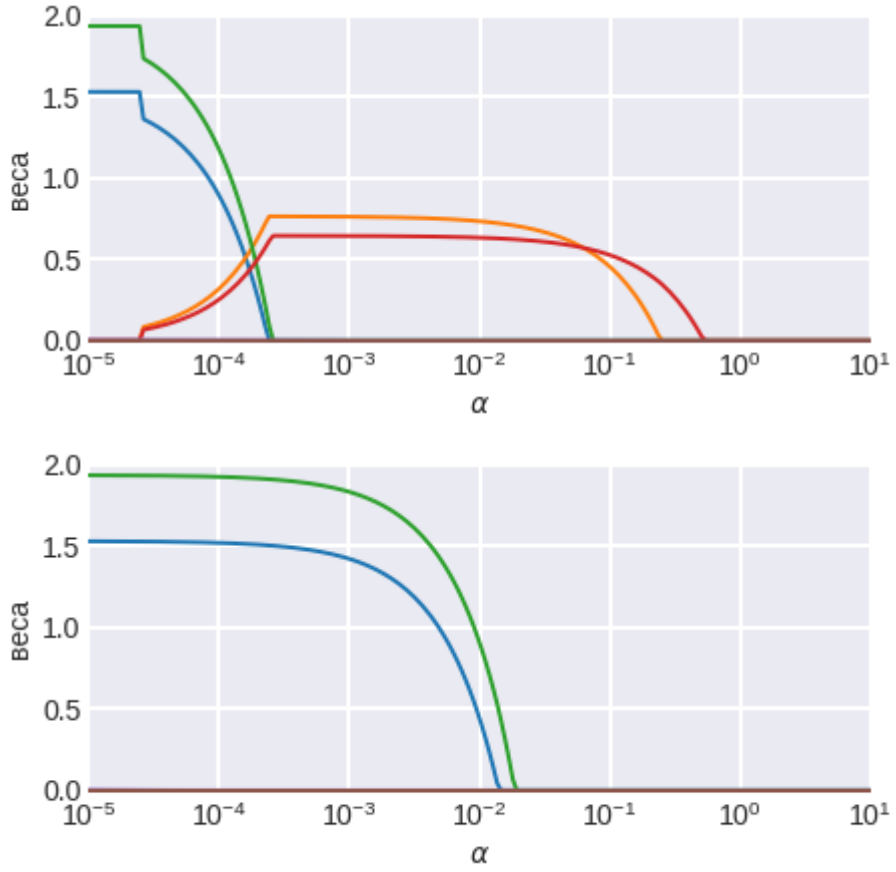


Рис. XX.8. Коэффициенты LASSO в исходной задаче (сверху), при использовании свободного члена (середина), при использовании свободного члена и стандартизации признаков (внизу).

Кроме описанных способов регуляризации при настройке линейных моделей, **Ridge**

$$|| y - Xw ||_2^2 + \lambda || w ||_2^2 \to \min_w$$

и **LASSO (Least Absolute Shrinkage and Selection Operator)**

$$|| y - Xw ||_2^2 + \lambda || w ||_1 \to \min_w$$

есть **эластичная сеть (Elastic Net)** – в некотором смысле, комбинация предыдущих:

$$|| y - Xw ||_2^2 + \lambda_1 || w ||_1 + \lambda_2 || w ||_2^2 \to \min_w,$$

в ней используется L1 и L2 регуляризации, каждая со своим весом. Геометрический смысл Ridge, LASSO и Elastic Net показаны на рис. XX.9. Если рассмотреть задачи оптимизации, которые соответствуют этим методам в постановке с ограничениями

$$\sum_{i=1}^{m}\left(y_i - w_0 - \sum_{j=1}^{n} w_j x_{ij}\right)^2 \to \min_w, \quad \sum_{j=1}^{n} w_j^2 \le \lambda,$$

$$\sum_{i=1}^{m}\left(y_i - w_0 - \sum_{j=1}^{n} w_j x_{ij}\right)^2 \to \min_w, \quad \sum_{j=1}^{n} |w_j| \le \lambda,$$

то видно, что в точках касаний линий ограничений и уровней оптимизируемых функций находится точка искомого минимума. Во втором случае касание произошло в вершине ромба. А как раз в его вершинах происходит зануления координат, т.к. они находятся на прямых с уравнениями $w_1 = 0$, $w_2 = 0$.

Теперь интуитивно понятно, почему при использовании L1-нормы в регуляризации возникает **«эффект разреженности»**, т.е. коэффициенты при некоторых признаках зануляются. Если случайно выбирать центр линий уровня оптимизируемой функции и их вытянутость, то при L1-ограничениях больше вероятность того, что касание произойдёт в точках с нулевыми координатами[1] (на рис. XX.9 показан случай конкретной задачи оптимизации).

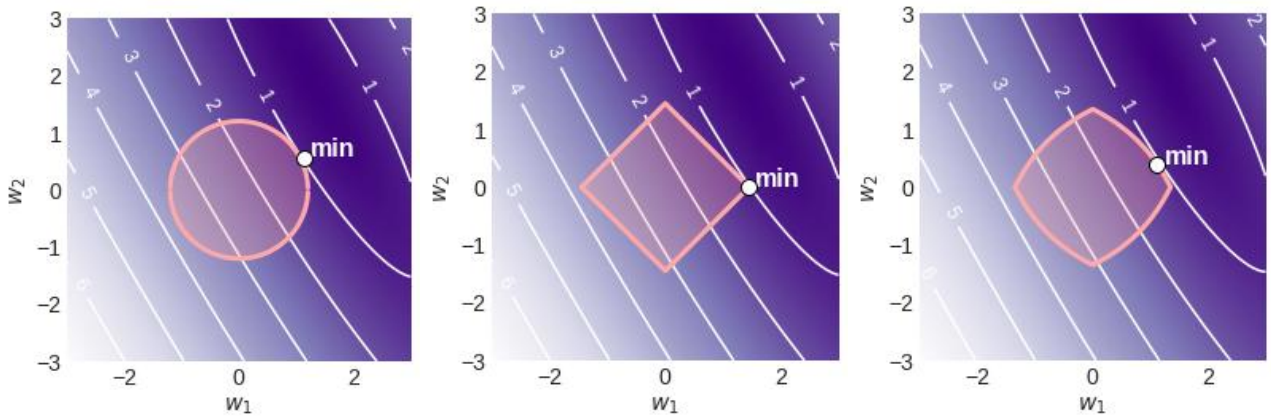


Рис. XX.9. Линии уровня и ограничения в Ridge (слева), LASSO (посередине) и Elastic Net (справа).

Другое интуитивное объяснение заключается в том, что L1-норма больше похожа на L0-норму, чем L2-норма, см. рис. XX.10, L0-норма равна числу ненулевых координат вектора:

$$\|w\|_0 = |\{t \mid w_t \ne 0\}|,$$

[1] Интересные иллюстрации этой вероятности можно найти здесь: David S. Rosenberg «Foundations of Machine Learning» https://bloomberg.github.io/foml/

поэтому она идеально подходит для отбора признаков (выбора среди множества признаков наиболее релевантных), но на практике не используется по причине сложности соответствующей оптимизации.

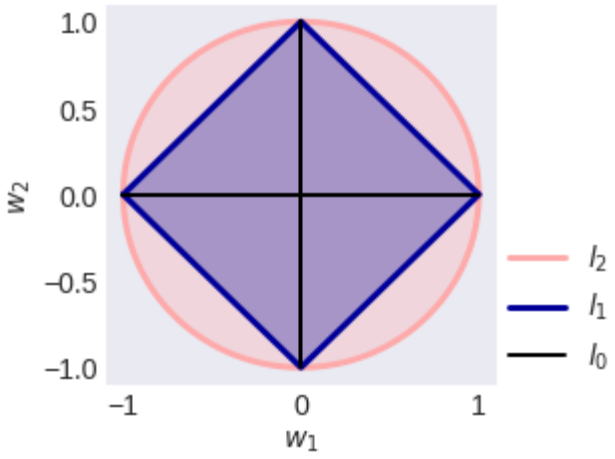


Рис. XX.10. Векторы ограниченной L0, L1 и L2 нормы.

Итак, при увеличении коэффициента регуляризации веса стремятся к нулю. Это приводит к некоторому упрощению[1] модели (соблюдению принципа Оккама), в случае LASSO – автоматической селекции признаков.

Кратко поговорим про второе решение проблемы вырожденности матрицы X^TX – **селекцию (отбор) признаков** (подробнее см. в соответствующей главе). Необходимость отбора можно проиллюстрировать примером линейно зависимых признаков, из-за которых наша матрица вырождена, если убрать лишние зависимости, оставив в матрице X только линейно-независимые столбцы, то при $m \geq n+1$ матрица X^TX станет невырожденной.

Заметим, что в методе LASSO автоматически произошёл отбор признаков: зануление весов означает, что решение не зависит от соответствующих признаков. Дальше мы увидим, что автоматический отбор признаков бывает и в других моделях (не только линейных).

Третье решение проблемы вырожденности матрицы X^TX – **уменьшение (сокращение) размерности признакового пространства** (это тоже тема отдельной главы). Пока приведём иллюстрацию, что такое сокращение размерности и чем это отличается от селекции признаков, см. рис. XX.11 – при сокращении размерности мы получаем новое признаковое пространство с меньшим число признаков, нет гарантии что новые признаки совпадают с

[1] В целом, конечно, неверно, что чем меньше параметров в модели, тем она проще. Но зануление веса признака в линейной модели точно не делает модель сложнее.

исходными, в общем случае они лишь как-то выражаются через них. Обоснование необходимости сокращения размерности такое же как и для селекции.

	x1	x2	x3	y
0	0.44	0.62	0.51	-0.25
1	0.03	0.53	0.07	-0.51
2	0.55	0.13	0.43	0.41
3	0.44	0.51	0.10	0.04
4	0.42	0.18	0.13	0.12
5	0.33	0.79	0.60	-0.45

	x1-x2	y
0	-0.18	-0.25
1	-0.50	-0.51
2	0.42	0.41
3	-0.07	0.04
4	0.24	0.12
5	-0.46	-0.45

Рис. XX.11. Пример сокращения размерности.

Последний способ решения проблемы вырожденности матрицы $X^{\mathrm{T}}X$ – **увеличение выборки** можно проиллюстрировать следующим образом. Если число объектов мало: $m \le n$, то матрица $X^{\mathrm{T}}X$ размера $(n+1)\times(n+1)$ заведомо вырождена, т.к.

$$\mathrm{rg}(X^{\mathrm{T}}X)_{(n+1)\times(n+1)} = \mathrm{rg}(X) \le \min(n+1, m) < n+1,$$

но если достаточно увеличить число объектов, то нарушится неравенство $m \le n$. Также при увеличении выборки могут устраниться какие-то линейные зависимости между столбцами.

Линейная регрессия: градиентный метод обучения

Мы уже сказали, что на практике не применяется аналитическое решение (XX.4). Вместо этого применяется метод стохастического градиентного спуска (Stochastic Gradient Descent, см. главу «Оптимизация»). Для оптимизации

$$\frac{1}{2}\sum_{i=1}^{m}(a(x_i \mid w) - y_i)^2 \to \min$$

итерационно уточняют параметры

$$w^{(t+1)} = w^{(t)} - \eta_t (a(x_i \mid w^{(t)}) - y_i)x_i,$$

где $a(x \mid w) = w^{\mathrm{T}} x$. На рис. XX.12-13 показаны изменения параметров в процессе такого уточнения и изменение ошибки.

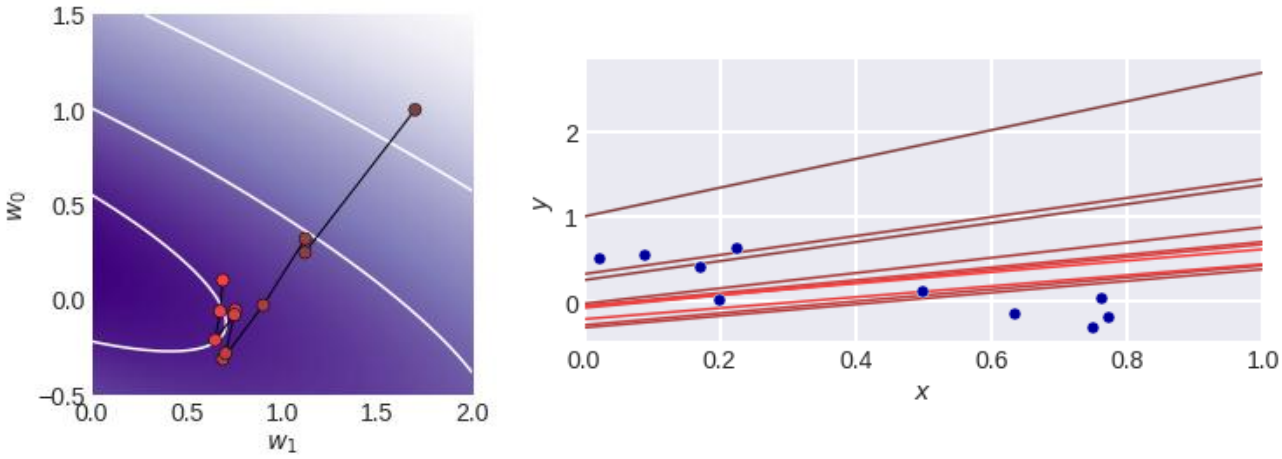


Рис. XX.12. Стохастический градиентный спуск в линейной регрессии: перемещения в пространстве параметров (слева), изменение прямой при этом перемещении (справа сверху).

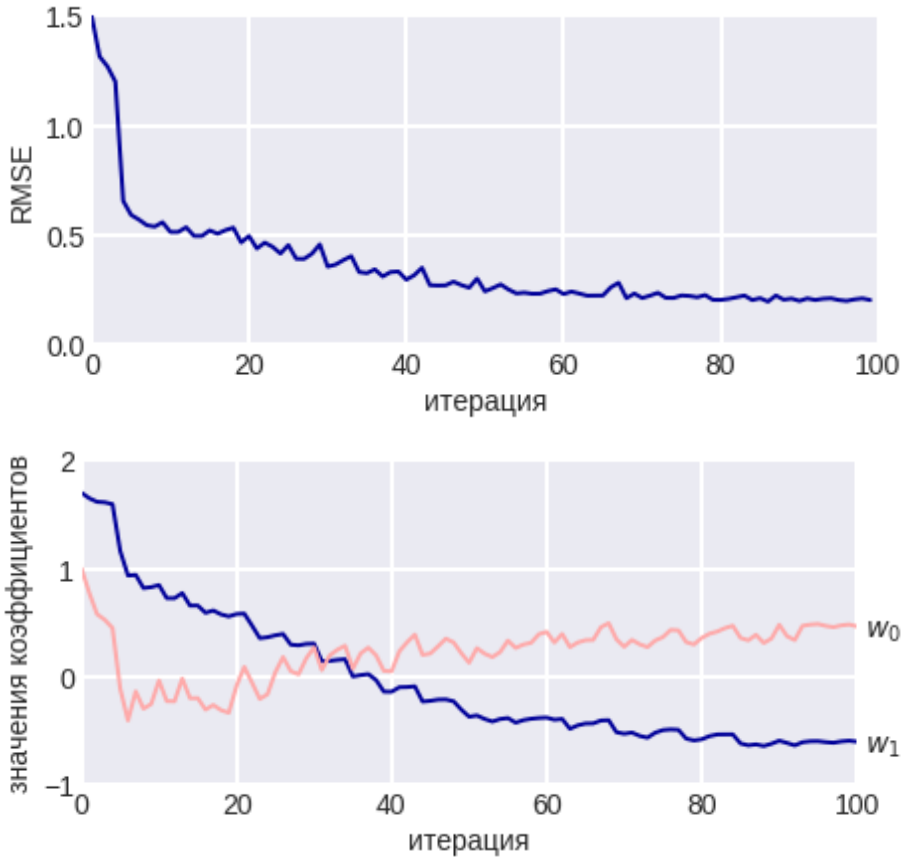


Рис. XX.13. Стохастический градиентный спуск в линейной регрессии: изменение ошибки (вверху) и изменение параметров (внизу).

Разные регрессии

На рис. XX.14 в нескольких задачах разным цветом показаны разные линейные регрессии. Как они получены и почему они разные? В обоих случаях используется линейная регрессия, но из двух признаков разные объявляются целевыми[1]. Синий график – решение ищется в виде $y(x): Y = w_0 + w_1 X_1$,

$$\left\| \begin{bmatrix} x_1 & 1 \\ \cdots & \cdots \\ x_m & 1 \end{bmatrix} \begin{pmatrix} w_1 \\ w_0 \end{pmatrix} - \begin{pmatrix} y_1 \\ \cdots \\ y_m \end{pmatrix} \right\|_2^2 \to \min,$$

розовый график – в виде $x(y): X_1 = w_0 + w_1 Y$,

$$\left\| \begin{bmatrix} y_1 & 1 \\ \cdots & \cdots \\ y_m & 1 \end{bmatrix} \begin{pmatrix} w_1 \\ w_0 \end{pmatrix} - \begin{pmatrix} x_1 \\ \cdots \\ x_m \end{pmatrix} \right\|_2^2 \to \min.$$

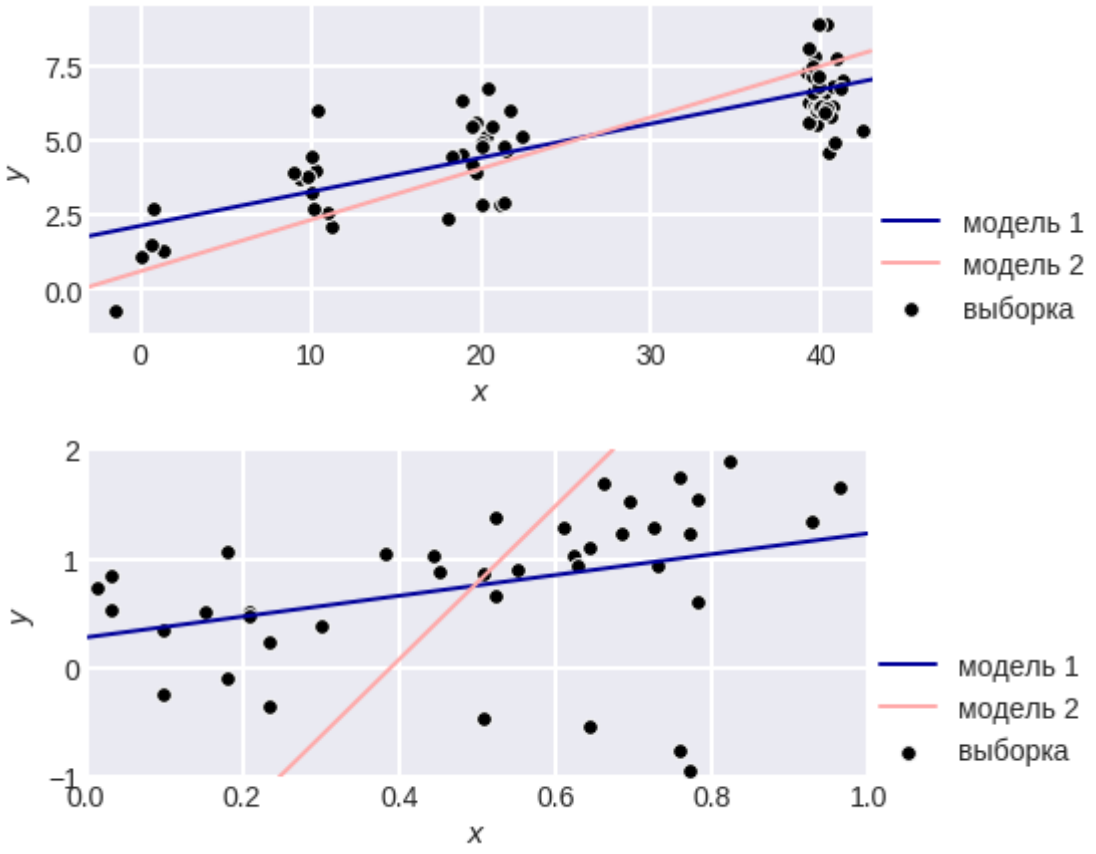


Рис. XX.14. Иллюстрация разных линейных регрессий в двух задачах.

[1] Есть и третья линейная регрессия, «промежуточная стратегия» – метод главных компонент (PCA), подробнее в главе об обучении на неразмеченных данных.

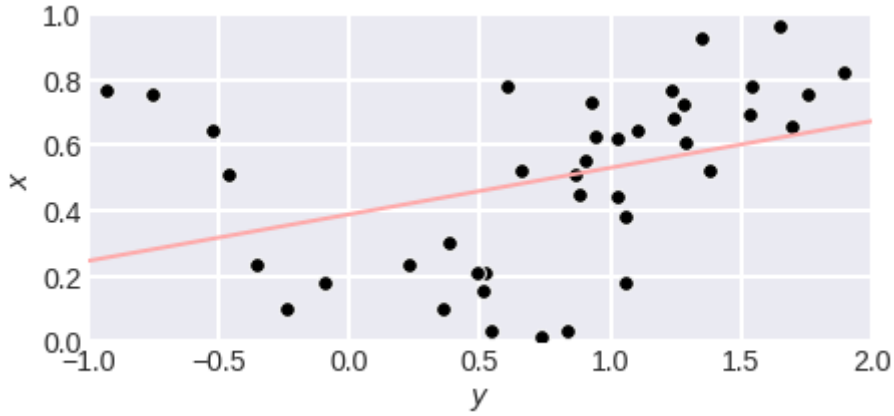


Рис. XX.15. Показана конфигурации нижнего рис. XX.14 в другой системе координат (оси поменяли местами).

Визуально кажется, что на рис. XX.14 синие линии точнее описывают линейную зависимость, но это оптический обман. Если поменять оси местами, см. рис. XX.15, то видно, что розовая линия также вполне «подогнана под данные» (другое дело, что зависимость здесь не линейная).

Неустойчивость к выбросам и робастная регрессия

Проиллюстрируем, что линейная регрессия неустойчива к выбросам, см. рис. XX.16 – здесь показана конфигурация точек, в которой явно выделяется одна, она расположена в стороне, такие точки называются выбросами. Показана линейная регрессия (соответствующая ей прямая), обученная по всей выборке и по выборкам в которых удалили по одному объекту. При удалении выброса решение существенно меняется, поэтому важно избавляться от подобных выбросов или строить **робастную регрессию** – устойчивую к подобным выбросам.

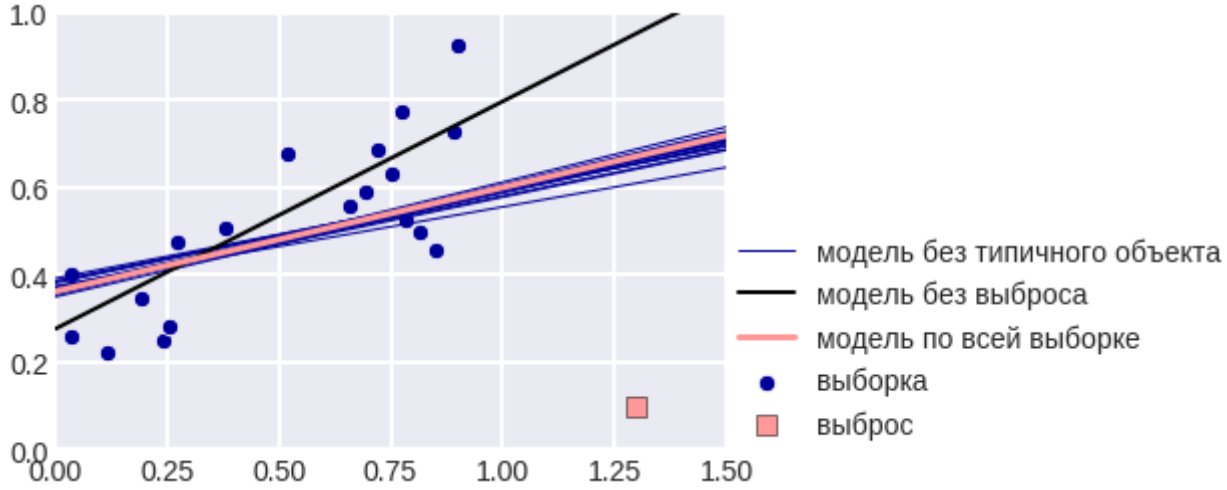


Рис. XX.16. Линейные регрессии построенные по разным выборкам.

Для построения робастной регрессии нам понадобится **регрессия с весами объектов**, в которой параметры регрессии ищутся в виде решения такой задачи оптимизации

$$\sum_{i=1}^{m} v_i^2 \left(y_i - w^{\mathrm{T}} x_i \right)^2 + R(w) \to \min, \qquad \text{(XX.6)}$$

где $R(w)$ – регуляризационная добавка (её может и не быть), $v_i^2 \in \mathbb{R}^+$ – цена ошибки на i-м объекте (квадрат использован для удобства, т.к. это неотрицательная величина). Заметим, что задачу можно переписать в виде

$$\sum_{i=1}^{m} \left(v_i y_i - w^{\mathrm{T}} (v_i x_i) \right)^2 + R(w) \to \min.$$

Получается, что регрессия с весами эквивалентна обычной регрессии, но с другой обучающей выборкой:

$$\{(x_1, y_1), \ldots, (x_m, y_m)\} \to \{(v_1 x_1, v_1 y_1), \ldots, (v_m x_m, v_m y_m)\}.$$

Также отметим, что описанный переход в матричном виде выглядит так (без регуляризации):

$$(y - Xw)^{\mathrm{T}} V^{\mathrm{T}} V (y - Xw) = \| Vy - VXw \|_2^2 \to \min_w,$$

здесь в диагональной матрице $V = \mathrm{diag}(v_1, \ldots, v_m)$ на диагонали стоят веса $v_1, \ldots, v_m$. Аналитическое решение в полученной задаче:

$$w = (X^{\mathrm{T}} V^{\mathrm{T}} V X)^{-1} X^{\mathrm{T}} V^{\mathrm{T}} V y.$$

В общем случае при решении задачи с весами используют также следующие приёмы:

1) переходят к новым данным (как было показано выше): признаки и целевые значения умножают на веса $v_1,\ldots,v_m$ (отметим, что мы существенно использовали линейность модели – для других моделей этот приём не пройдёт, кроме того, неверно говорить, что здесь просто все признаки умножаются на вес объекта, так в регрессии со свободном членом фиктивный единичный признак также умножится);

Не забывайте про фиктивный признак: он часто «ломает» очевидную интерпретацию некоторых преобразований.

2) если $v_1^2,\ldots,v_m^2$ целые числа, то можно продублировать объекты: внести соответствующее число копий каждого объекта в выборку, формально это приводит к эмпирическому риску (XX.6);

3) использовать стохастический градиентный спуск (SGD) со специальными вероятностями выбора объектов. Это более универсальная техника, подходит для алгоритмов, которые обучаются с помощью SGD, а также для алгоритмов при обучении которых происходит сэмплирование объектов (взятие подвыборок). Например, в методе SGD i -й объект выбирается с вероятностью

$$\frac{v_i^2}{v_1^2+\cdots+v_m^2}.$$

Опишем **устойчивую к шумовым объектам регрессию (Robust Regression)**:

0. Инициализация весов объектов: $v=(v_1,\ldots,v_m)=(1/m,\ldots,1/m)$.

1. Цикл.

1.1. Настроить алгоритм, учитывая веса объектов: $a=\text{fit}(\{x_i,y_i,v_i\})$.

1.2. Вычислить ошибки на обучении $\varepsilon_i=a(x_i)-y_i$.

1.3. Пересчитать веса объектов $v_i=\exp(-\gamma\varepsilon_i^2)$ (можно использовать другую невозрастающую на $[0,+\infty)$ функцию, иногда веса нормируют – делят на сумму весов по всем объектам).

Обратим внимание, что устойчивую регрессию можно делать на базе любой регрессионной модели. На рис. XX.17 показано, что на разных объектах совершаются разные ошибки, чтобы выбросы не оказывали большого влияния на модель надо чтобы их веса были небольшими. Поэтому веса объектов тем

больше, чем меньше ошибка на них. В описанном методе веса и параметры модели итерационно уточняются до сходимости. Гиперпараметр $\gamma \in \mathbb{R}^+$ в формуле для весов можно выбрать с помощью скользящего контроля (см. главу «Контроль»).

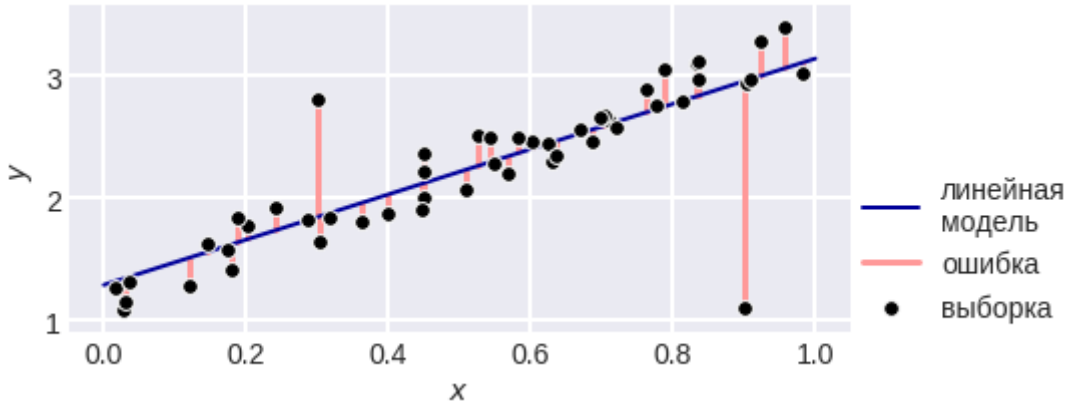


Рис. XX.17. Линейная регрессия и ошибки на разных объектах.

Другой устойчивый алгоритм **RANdom SAmple Consensus (RANSAC)** заключается в проведении следующих шагов:

1. Цикл

    1.1. Выбрать случайное подмножество k точек – базовое (inliers).

    1.2. Обучить модель на базовом подмножестве.

    1.3. Найти все объекты, на которых модель «хорошо работает», например, ошибка не превышает заранее заданный порог ε.

    1.4. Пополнить ими базовое множество.

    1.5. (опционально) Обучить модель на новом базовом множестве (возможно также удалить из него объекты с большой ошибкой – больше ε).

2. Выбрать модель с наибольшим базовым множеством (при равенстве мощностей базовых множеств учитывается ошибка на нём / на всей выборке).

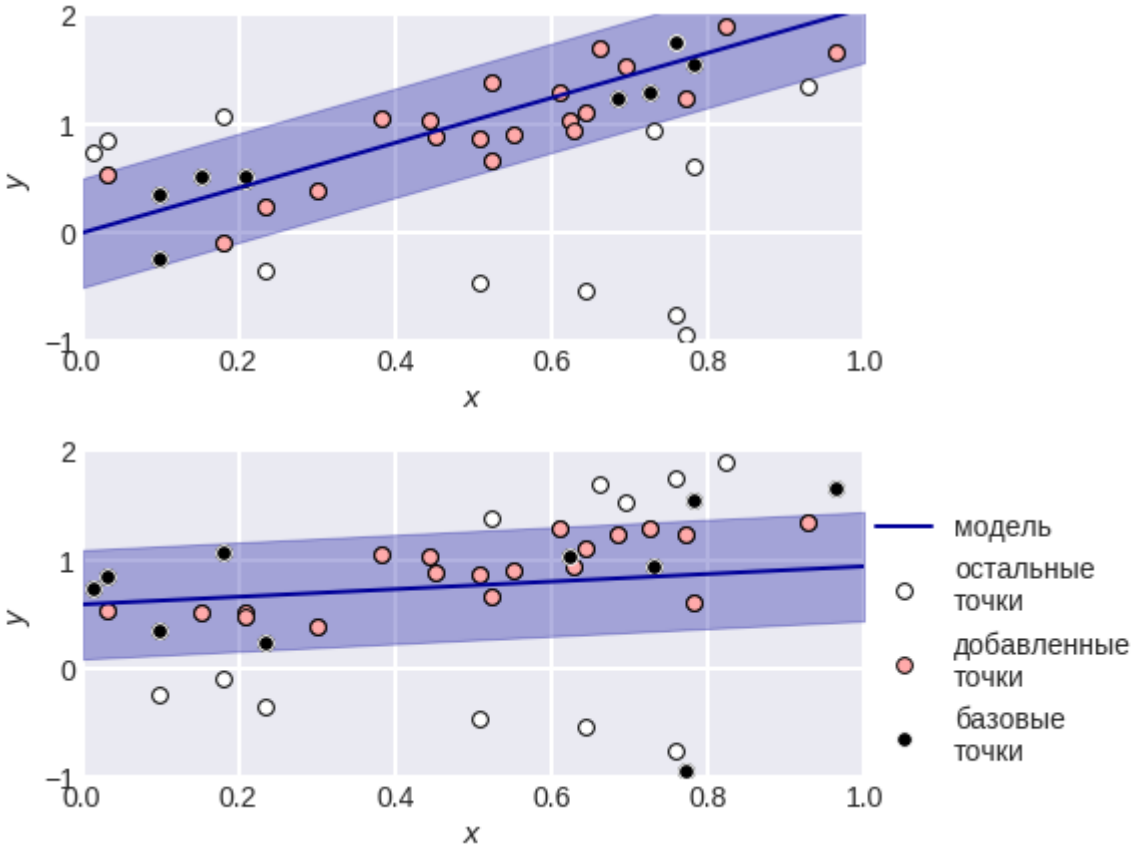


Рис. XX.18. Две итерации RANSAC.

На рис. XX.18 показаны две итерации цикла: выбраны два разных базовых множества (точки чёрного цвета), в полосе находятся точки, на которых модель хорошо работает (розовым цветом). В указанной конфигурации точек просматриваются два паттерна: две линейных зависимости, одна представлена большим числом объектов. Именно её и должен в итоге обнаружить алгоритм RANSAC: при случайном выборе базового множества есть большая вероятность, что все его представители (или подавляющая часть) будут описывать «большую» закономерность. Результат представлен на рис. XX.19.

Обратим также внимание, что в качестве базового регрессионного алгоритма в методе RANSAC может использоваться любой (не обязательно линейный). Число базовых точек k придётся выбрать экспертно.

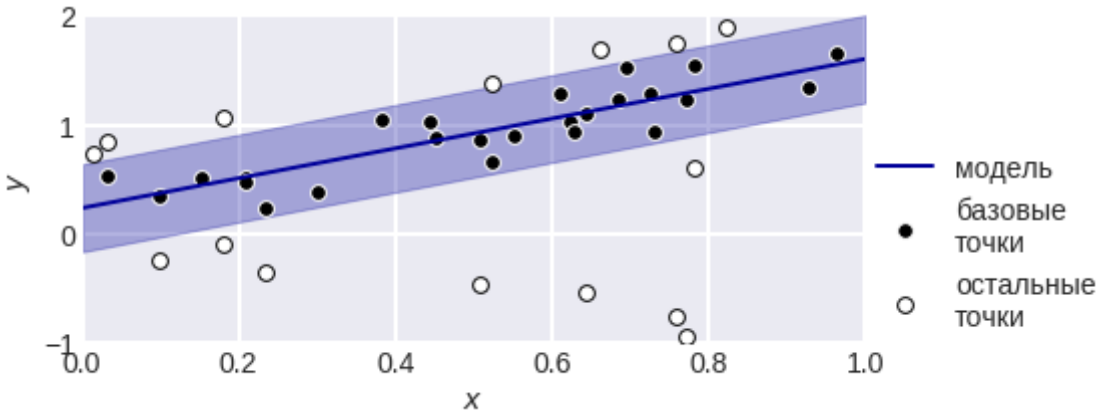


Рис. XX.19. Результат алгоритма RANSAC – полученная модель и её базовое множество.

Ошибка и её оценка (невязка)

Вернёмся к линейной регрессии: пусть есть линейная зависимость с точностью до шума

$$y = Xw^* + \varepsilon$$

(запись в векторном виде), при решении этой задачи с помощью линейной регрессии мы на исходной выборке получаем ответы

$$a = Xw$$

w – это найденные нами параметры линейной регрессии, мы получили для них явную формулу

$$w = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y,$$

т.е. наш вектор невязки

$$\hat{\varepsilon} = y - a = y - Xw = y - X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y = (I - H)y =$$

$$= (I - H)(Xw^* + \varepsilon) = Xw^* - HXw^* + (I - H)\varepsilon =$$

$$= Xw^* - X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Xw^* + (I - H)\varepsilon = (I - H)\varepsilon,$$

здесь мы ввели обозначение $H = X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}$ (выделено синим) эта матрица называется **projection (hat) matrix**, красная часть равна единичной матрице. Термин проекционная связан с тем, что

$$a = Hy \in L(X),$$

т.е. вектор a лежит в линейной оболочке столбцов матрицы X, т.к. эта матрица умножается справа на некоторый вектор:

$$Hy = X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y.$$

(вектор получается после вычисления выражения, выделенного синим). На рис. XX.20 показано как матрица H проецирует на пространство $L(X)$ вектор y. Кстати,

$$H^2 = X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}} = X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}} = H,$$

т.е. повторная проекция не меняет вектор. Можно убедиться (попробуйте) в том, что эта проекция ортогональная.

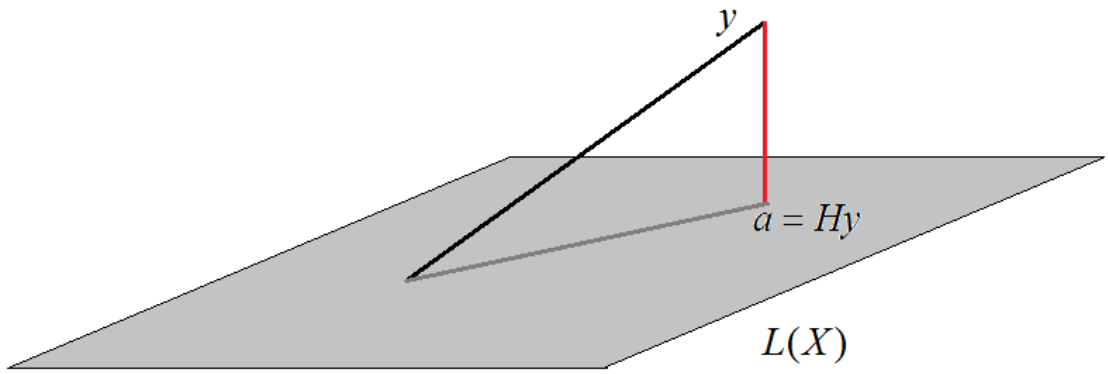


Рис. XX.20. Проекция вектора y на пространство $L(X)$.

На рис. 21 показана модельная задача и значения проекционной матрицы в ней (при этом точки в выборке упорядочены по возрастанию единственного признака).

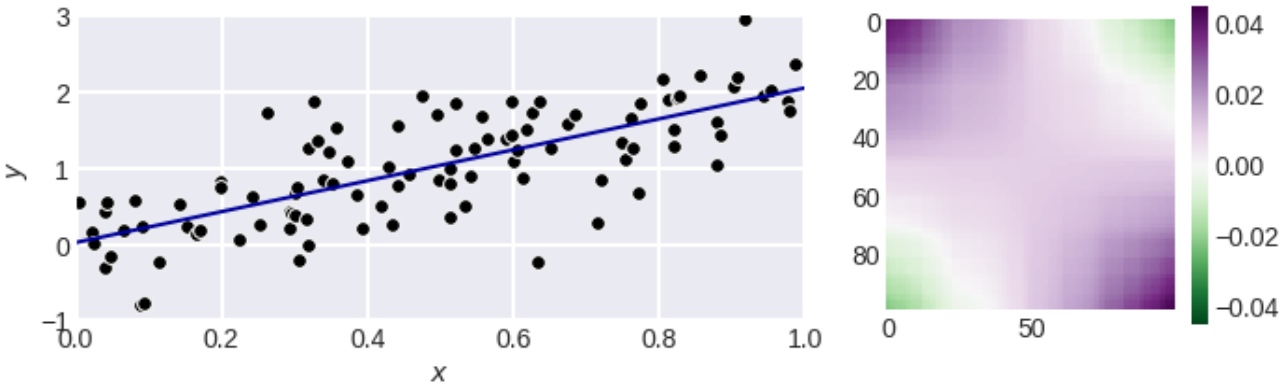


Рис. XX.21. Задача регрессии (слева) и значения проекционной матрицы (справа).

Итак, мы получили, что $\hat{\varepsilon} = (I - H)\varepsilon$, отсюда можно сделать несколько выводов:

- невязки коррелируют (даже если «истинные ошибки» ε были независимы),
- но у невязок нет корреляции с целевым значением (если её не было у ошибок ε), заметим что в формуле вообще нет вектора y,
- для справки: во многих статистических пакетах невязки стандартизуют по формуле

$$\hat{\varepsilon}_{[t]} / \sqrt{1 - h_{tt}}$$

(справедливости ради отметим, что это даёт незначительный эффект).

Для диагональных элементов проекционной матрицы может быть получено явная формула

$$h_{tt} = \frac{1}{m} + \frac{(x_t - \overline{x})^2}{\sum_{i=1}^{m} (x_i - \overline{x})^2},$$

эти значения связаны с т.н. **расстоянием Кука (Cook's distance)**

$$D_j = \text{const} \cdot \sum_{i=1}^{m} (a(x_i \mid X_{\text{train}}) - a(x_i \mid X_{\text{train}} \setminus \{x_j\}))^2,$$

которое показывает насколько j-й объект влияет на модель (здесь разница в ответах на объектах выборки модели, которая обучена на всех данных, и модели, которая обучена на всех данных без j-го объекта). Оказывается, что

$$D_j = \text{const} \cdot \frac{h_{jj}}{(1 - h_{jj})^2} \hat{\varepsilon}_{[j]}^2,$$

т.е. расстояние Кука зависит от диагонального элемента проекционной матрицы и величины ошибки. На рис. XX.22 показана степень влияния объекта в зависимости от его координаты в рассматриваемой модельной задаче. У нас все объекты находятся на отрезке $[0,1]$, для каждого можно вычислить соответствующий диагональный элемент h_{jj} и выражение, которое входит в расстояние Кука $h_{jj} / (1 - h_{jj})^2$ (мы видим, что эти значения похожи). Поскольку

все объекты довольно плотно замощают отрезок $[0,1]$, по точкам (x_j, h_{jj}) можно построить график –см. рис. XX.22.

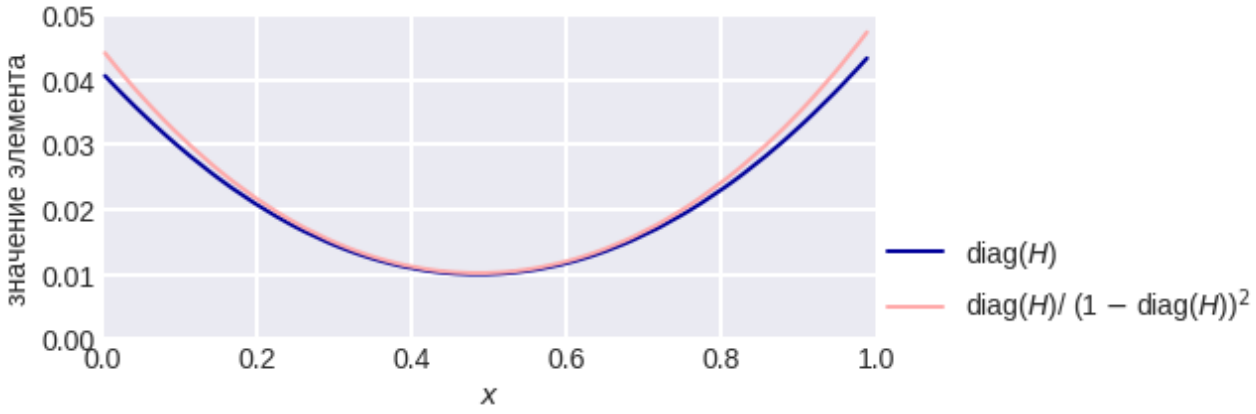


Рис. XX.22. Степень влияния объектов на модель.

Видно, что при одинаковых ошибках сильнее на модель влияют объекты, расположенные на краю отрезка (в центре влияние в несколько раз меньше).

Линейная регрессия: связь с сингулярным разложением (SVD)

Рассмотрим гребневую регрессию (в формулах красным цветом будет отмечать, что добавляется при регуляризации): оптимальные параметры регрессии

$$w = (X^{\mathrm{T}} X + \lambda I)^{-1} X^{\mathrm{T}} y,$$

воспользуемся полным сингулярным разложением (SVD) матрицы X на матрицы размеров $(m \times m) \cdot (m \times n) \cdot (n \times n)$:

$$X = U \Lambda V^{\mathrm{T}},$$

тогда (пользуясь ортогональностью матриц U, V)

$$w = (V \Lambda^{\mathrm{T}} U^{\mathrm{T}} U \Lambda V^{\mathrm{T}} + \lambda I)^{-1} V \Lambda^{\mathrm{T}} U^{\mathrm{T}} y = (V \Lambda^{\mathrm{T}} \Lambda V^{\mathrm{T}} + \lambda V V^{\mathrm{T}})^{-1} V \Lambda^{\mathrm{T}} U^{\mathrm{T}} y =$$

$$= (V(\Lambda^{\mathrm{T}} \Lambda + \lambda I) V^{\mathrm{T}})^{-1} V \Lambda^{\mathrm{T}} U^{\mathrm{T}} y =$$

$$= V(\Lambda^{\mathrm{T}} \Lambda + \lambda I)^{-1} \Lambda^{\mathrm{T}} U^{\mathrm{T}} y,$$

где $k = \min(m,n)$. Заметим, что в середине находится матрица, которая имеет вид

$$(\Lambda^{\mathrm{T}} \Lambda + \lambda I)^{-1} \Lambda^{\mathrm{T}} = \mathrm{diag}(\lambda_1 / (\lambda_1^2 + \lambda), \ldots, \lambda_k / (\lambda_k^2 + \lambda))_{n \times m},$$

если это не очевидно, приведём несколько примеров:

$$\left(\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\end{bmatrix}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\\ 0 & 0\end{bmatrix}+\lambda\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\right)^{-1}\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\end{bmatrix}=$$

$$=\left(\begin{bmatrix}\lambda_1^2+\lambda & 0\\ 0 & \lambda_2^2+\lambda\end{bmatrix}\right)^{-1}\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\end{bmatrix}=\begin{bmatrix}\frac{\lambda_1}{\lambda_1^2+\lambda} & 0 & 0\\ 0 & \frac{\lambda_2}{\lambda_1^2+\lambda} & 0\end{bmatrix},$$

$$\left(\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\\ 0 & 0\end{bmatrix}\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\end{bmatrix}+\lambda\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\right)^{-1}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\\ 0 & 0\end{bmatrix}=$$

$$=\left(\begin{bmatrix}\lambda_1^2+\lambda & 0 & 0\\ 0 & \lambda_2^2+\lambda & 0\\ 0 & 0 & +\lambda\end{bmatrix}\right)^{-1}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\\ 0 & 0\end{bmatrix}=\begin{bmatrix}\frac{\lambda_1}{\lambda_1^2+\lambda} & 0\\ 0 & \frac{\lambda_2}{\lambda_1^2+\lambda}\\ 0 & 0\end{bmatrix}.$$

Таким образом,

$$w=\sum_{j=1}^{k}\frac{\lambda_j\cdot u_j^{\mathrm{T}}y}{\lambda_j^2+\lambda}v_j\,,$$

здесь v_j – j-й столбец матрицы V, u_j – j-й столбец матрицы U. Получается, что вектор параметров w – линейная комбинация столбцов V, коэффициенты в линейной комбинации – скалярные произведения столбцов U и целевого столбца, с нормировочными коэффициентами, зависящими от сингулярных чисел и коэффициента регуляризации. Виден эффект от регуляризации: при больших λ зануляются коэффициенты, без регуляризации, когда $\lambda_j\approx 0$, коэффициенты могут быть большими!

Кстати, число обусловленности матрицы $(X^{\mathrm{T}}X+\lambda I)$

$$\frac{\lambda_{\max}(X^{\mathrm{T}}X+\lambda I)}{\lambda_{\min}(X^{\mathrm{T}}X+\lambda I)}=\frac{\lambda_{\max}^2+\lambda}{\lambda_{\min}^2+\lambda}\xrightarrow[\lambda\to+\infty]{}1,$$

поэтому при регуляризации указанная матрица становится хорошо обусловленной.

Посмотрим теперь на проекционную матрицу $H = X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}$:

$$H = U\Lambda V^{\mathrm{T}}(V\Lambda^{\mathrm{T}}U^{\mathrm{T}}U\Lambda V^{\mathrm{T}})^{-1}V\Lambda^{\mathrm{T}}U^{\mathrm{T}} =$$

$$= U\Lambda V^{\mathrm{T}}(V\Lambda^{\mathrm{T}}\Lambda V^{\mathrm{T}})^{-1}V\Lambda^{\mathrm{T}}U^{\mathrm{T}} =$$

$$= U\Lambda V^{\mathrm{T}}V(\Lambda^{\mathrm{T}}\Lambda)^{-1}V^{\mathrm{T}}V\Lambda^{\mathrm{T}}U^{\mathrm{T}} =$$

$$= U\Lambda(\Lambda^{\mathrm{T}}\Lambda)^{-1}\Lambda^{\mathrm{T}}U^{\mathrm{T}}$$

И тут тонкость в том, что зелёная матрица может не быть единичной, приведём пример, когда это так:

$$\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\\ 0 & 0\end{bmatrix}\left(\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\end{bmatrix}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\\ 0 & 0\end{bmatrix}\right)^{-1}\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}.$$

Поиск ортогонального соответствия (Orthogonal matching pursuit)

Пусть X_J – матрица, образованная столбцами из множества $J \cup \{0\}$, $J \subseteq \{1,2,\ldots,n\}$ (под столбцом с нулевым номером понимаем столбец, состоящий из единиц, соответствующий фиктивному признаку). Изначально $J = \varnothing$ (т.е. в X_J входит лишь нулевой столбец), на каждой итерации решаем задачу

$$\| y - X_{J\cup\{j\}}w\|_2^2 \to \min_{w,j}$$

и наращиваем множество используемых признаков

$$J \leftarrow J \cup \{j\}$$

пока $\| y - X_J w\|_2^2 \geq \varepsilon$ или $|J| < k$. Таким образом, мы постепенно наращиваем множество столбцов, каждый следующий выбирая таким образом, чтобы минимизировать суммарную ошибку. Останавливаемся, когда набрали достаточное число признаком или ошибка достаточно уменьшилась. Эту идею (последовательного наращивания признаков) можно применять и для других семейств алгоритмов (не только для линейных).

Пусть изначально из матрицы удалены столбцы-дубликаты, все столбцы пронормированы, тогда вместо

$$\| y - X_{J\cup\{j\}} w \|_2^2 \to \min_{w,j}$$

решаем такую последовательность задач

$$j = \arg\max_{j\notin J} (X[:,j]^{\mathrm{T}} (y - X_J w)),$$

$$J \leftarrow J \cup \{j\},$$

$$w \leftarrow \arg\min_{w} \| y - X_J w \|_2^2.$$

Здесь через $X[:,j]$ обозначен j-й столбец матрицы объект-признак, из-за предварительной нормировки задача максимизации эквивалентна поиску столбца, который максимально коррелирует с текущим вектором ошибок.

Приложения линейных методов

Часто линейные методы можно успешно применять **в задачах с текстами**, когда для их представления используется «мешок слов» (bag of words, BoW[1]). Например, автор успешно использовал линейную регрессию в задаче соревнования «Topical Classification of Biomedical Research Papers[2]». В нём матрица документ-слово размера 10000×25000 была разложена с помощью неполного сингулярного разложения:

$$X_{q\times n} \approx U_{q\times k} L_{k\times k} V_{k\times n}$$

и первая матрица разложений использовалась как признаковая. Число компонент k в разложении было гиперпараметром метода и оптимизировалось перебором. Необходимость в разложении вызвано желанием сократить число признаков (и настраиваемых параметров). Дальше использовалась многомерная линейная регрессия:

[1] В BoW каждый текст задаётся вектором чисел вхождений слов из словаря (некоторого заранее заданного и фиксированного перечня слов), а корпусу текстов соответствует матрица размера «число документов» × «число слов в словаре».

[2] Janusz A. et al. JRS'2012 data mining competition: Topical classification of biomedical research papers //International Conference on Rough Sets and Current Trends in Computing. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2012. – С. 422-431.

$$||U_{q\times k}W_{k\times l} - Y_{q\times l}||_2^2 \to \min_W,$$

здесь матрица Y – бинарная матрица-ответ, её ij-й элемент равен единице тогда и только тогда, когда i-й документ принадлежит j-му классу (одновременно документ может принадлежать нескольким классам). После применения линейной регрессии подбирался некоторый порог и бинаризацией по этому порогу (bin) получалась окончательная классификация:

$$\text{bin}(UW).$$

Линейная регрессия успешно использовалась **в задаче прогнозирования дебита нефти** на платформе boosters.pro. Фактически решалась задача прогнозирования временного ряда линейным методом с ограничениями на коэффициенты:

$$y_t = \sum_{i=0}^{k} w_{ti} y_{-i}, \quad w_{t0} \geq w_{t1} \geq \ldots,$$

при данном ряде $(\ldots, y_{-2}, y_{-1}, y_0)$ и необходимости предсказывать следующие значения $y_1, y_2, \ldots$. Ограничения и модель определялись экспертными знаниями, на рис. XX.23 показаны примеры рядов из соревнования, толстые участки – отрезки для прогнозирования, они описывают добычу нефти после смены насоса на скважине.

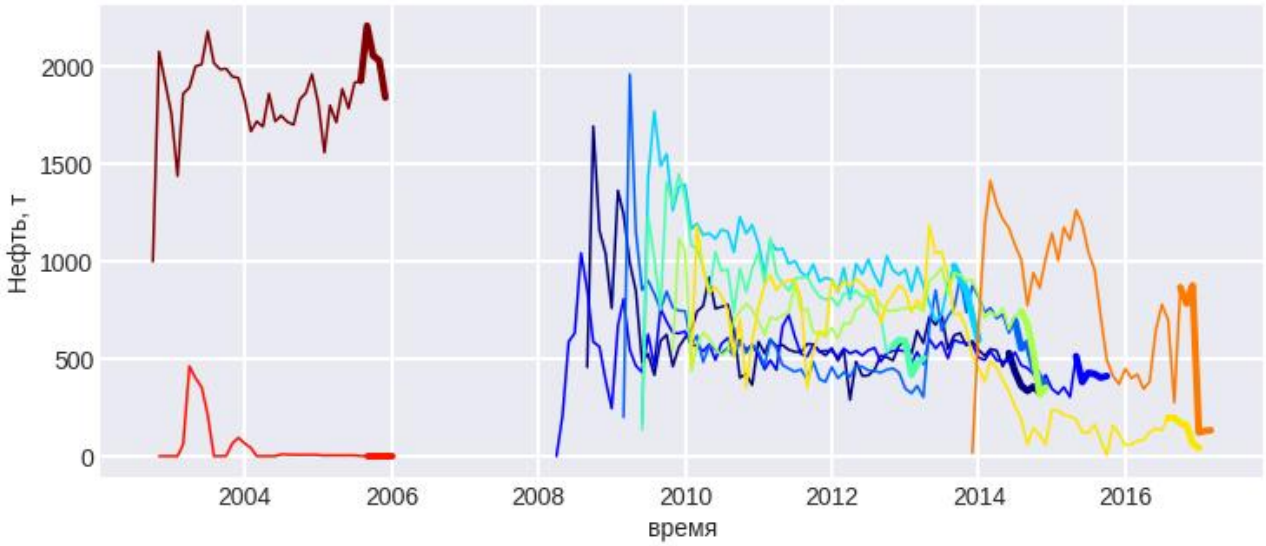


Рис. XX.23. Примеры рядов задачи прогнозирования дебита нефти.

При прогнозировании спроса (для каждого id товара предсказать, сколько его единиц будет куплено в течение некоторого периода времени) также неплохо

работают линейные методы, но лучше уже использовать простейшие нелинейности, например

$$Y = \max\left[\sum_t w_t X_t, 0\right],$$

а в качестве признаков можно брать:

- число покупок за последние k дней,
- число просмотров за последние k дней
- длина последнего непрерывного периода без покупок (в днях),
- изменение цены за последние k дней (в процентах),
- есть ли маркетинговая акция и т.п.

Линейные методы: промежуточные итоги

Из плюсов линейных методов можно отметить:

- простоту, надёжность, скорость работы, популярность,
- интерпретируемость (возможность с их помощью находить линейные закономерности в данных),
- возможность решать задачи интерполяции и экстраполяции (метрические алгоритмы и решающие деревья таким свойством не обладают),
- возможность добавления нелинейности с помощью генерации новых признаков (дальше расскажем про то, как это можно автоматизировать),
- методы хороши для теоретических исследований (в Ridge-регрессии есть явная формула для коэффициентов),
- коэффициенты асимптотически нормальны[1] (можно тестировать гипотезы о влиянии признаков),
- глобальный минимум в оптимизируемом функционале (при использовании регуляризации).

[1] Про это мы не рассказывали.

Из минусов линейных методов отметим, что

- линейная гипотеза вряд ли верна на практике,
- в теоретическом обосновании предполагается нормальность ошибок модели и использовании конкретной функции ошибки, см. главу «Функции ошибок»,
- классические линейные методы «страдают» из-за выбросов в обучающей выборке,
- в наиболее успешных примерах применения признаки однородны,
- есть проблема линейной зависимости признаков (необходимость регуляризации, селекции признаков или уменьшения размерности, увеличения объёма выборки).

Задачи

1. Перепишите уравнение прямой в линейной регрессии от одной переменной так, чтобы коэффициент, отвечающий за наклон зависел не от ковариации $\{x_i\}$ и $\{y_i\}$, а от корреляции (в явном виде).

2. Докажите, что пространства линейных комбинаций столбцов матриц X и $X^{\mathrm{T}}X$ совпадают. Верно ли это для матриц X и HX, где H – произвольная квадратная невырожденная матрица подходящего размера?

3. При наличии мультиколлинеарности возможна неверная интерпретация решения, например, от какого-то признака целевая зависимость должна быть монотонна, а его вес отрицательный. Приведите примеры, когда возникают такие эффекты. Есть ли гарантия их отсутствия при линейно независимых признаках?

4. При L1-регуляризации больше вероятность (чем при L2) того, что какие-то веса обнулятся. Можно ли вывести формулу для этой вероятности (как функцию от коэффициента регуляризации λ)? Может ли какой-то вес не обнулиться при сколь угодно большом значении λ? Верно ли, что в Elastic Net эта вероятность меньше, чем в LASSO, но больше, чем в Ridge?

5. Пусть необходимо искать в матрице данных $\| x_{ij} \|_{m \times n}$ линейные зависимости, т.е. такие наборы столбцов $J \subseteq \{1,2,\ldots,n\}$ и коэффициенты $\{w_j\}_{j \in J}$, не все равные нулю, что

$$\forall i \in \{1,2,\ldots,m\} \quad \sum_{j \in J} w_j x_{ij} = 0 .$$

Как лучше провести такой поиск? Как гарантированно найти все такие наборы столбцов? Обратите внимание, что в зашумлённых данных столбцы могут быть «почти линейно зависимыми», а также, что у нас нет целевого вектора.

6. Используя идею алгоритма RANSAC, предложите метод решения задачи при наличии такой целевой зависимости:

$$y_i = \begin{cases} w^{\mathrm{T}} x_i + \varepsilon_i, & \text{с вероятностью } p, \\ v^{\mathrm{T}} x_i + \varepsilon_i, & \text{иначе,} \end{cases}$$

где ε_i – шум (т.е. одновременно есть две линейные зависимости, в идеале надо найти векторы w и v).

7. Покажите (можно экспериментально), что случайная квадратная матрица невырождена с вероятностью 1. Зависит ли это от способа получения матрицы (от распределения её элементов)? Какие следствия из этого для применения линейной регрессии на практике?

Код к главе

```
from sklearn.linear_model import Ridge

model = Ridge(alpha=0.0) # ридж-регрессия
# обучение
model.fit(x_train[:, np.newaxis], y_train)
# обратите внимание: np.newaxis
# контроль
a_train = model.predict(x_train[:, np.newaxis])
a_test = model.predict(x_test[:, np.newaxis])
```

Код. Получение рис. XX.5.

```
X = np.random.rand(1000, 11)
X[:,1] = X[:,0]
X[:,4] = X[:,2] + X[:,3]
X[:,8] = X[:,5] + X[:,6] + X[:,7]
y = X[:,0] + 0.8*X[:,4] + 0.4*X[:,8] + 0.2*X[:,9] +
            0.5*np.random.randn(1000)
```

Код. Описанные выше графики XX.4 - XX.5 строились для такой задачи.

```
from sklearn.linear_model import RANSACRegressor
# Robustly fit linear model with RANSAC algorithm
ransac = RANSACRegressor()
ransac.fit(x[:, np.newaxis], y)
inlier_mask = ransac.inlier_mask_
outlier_mask = np.logical_not(inlier_mask)
```

Код. Использование алгоритма RANSAC.

Спасибо за внимание к книге!
Замечания по содержанию, замеченные ошибки и неточности можно написать в телеграм-чате
https://t.me/Dyakonovsbook